

# New Upper Bounds on The Approximability of 3D Strip Packing

Xin Han<sup>1</sup> Kazuo Iwama<sup>1</sup> Guochuan Zhang<sup>2</sup>

<sup>1</sup> School of Informatics, Kyoto University, Kyoto 606-8501, Japan  
 {hanxin, iwama}@kuis.kyoto-u.ac.jp

<sup>2</sup> Department of Mathematics, Zhejiang University, China  
 zgc@zju.edu.cn

## Abstract

In this paper, we study the 3D strip packing problem in which we are given a list of 3-dimensional boxes and required to pack all of them into a 3-dimensional strip with length 1 and width 1 and unlimited height to minimize the height used. Our results are below: i) we give an approximation algorithm with asymptotic worst-case ratio 1.69103, which improves the previous best bound of  $2 + \epsilon$  by Jansen and Solis-Oba of SODA 2006; ii) we also present an asymptotic PTAS for the case in which all items have *square* bases.

## 1 Introduction

For packing 2D items into bins or a strip, it is a natural idea to exploit techniques for packing lower dimensional (i.e., 1D) items. The *two-stage packing* is particularly well-known: As shown in Fig. 1 (a), a bin (or a strip) is divided into *shelves* and each shelf contains a single layer of items. After packing items into shelves, the problem of packing shelves into bins (or a strip) obviously becomes the 1D bin (or strip) packing problem. The idea originally comes from cutting a large board into smaller items efficiently [10]; one can cut the board only in two stages, i.e., cutting horizontally first and then vertically.

It should be noted that many existing 2D packing algorithms [5, 6, 3] are based on this two-stage packing. In 2002, Caprara [3] established the relation between 2D Bin Packing (2BP) and 2D Shelf Bin Packing (2SBP). Namely the maximum ratio between the optimal cost for 2SBP and that for 2BP is equal to  $T_\infty = 1.691\dots$  which is the well-known approximation factor of the Harmonic algorithm for 1D Bin Packing [19]. (A similar relation between 2D Strip Packing (2SP) and 2D Shelf Strip Packing (2SSP) was also established by Csirik and Woeginger [8].) As an important byproduct, Caprara also showed an approximation algorithm for 2BP whose asymptotic worst-case ratio is arbitrarily close to  $T_\infty$ , which first broke the barrier of two for the upper bound on the approximability of this problem.

**Our contribution** This paper extends the two-stage packing into the 3D Strip Packing (3SP) and obtains an approximation algorithm whose asymptotic worst-case ratio is arbitrarily close to  $T_\infty$ . Our model is standard (see Section 2 for details) and the previous best bound is  $2 + \epsilon$  by Jansen and Solis-Oba [14]. We also show that there is an APTAS for the special case in which all items have square bases.

Our algorithms use a *segment* as shown in Fig. 1 (b) instead of a shelf in the 2D case. For packing items (whose three sides are all at most 1.0) into a segment, we first divide a segment into slips and pack the items into slips by the next-fit (NF) algorithm. The key idea is to make the height  $c$  of each segment sufficiently large (within a constant), which effectively kills the inefficiency of the algorithm for the vertical direction in the sense that the unused space at the top of the segment is relatively small. After packing items into segments of the fixed height ( $=c$ ) and fixed length ( $=1.0$ ), we can obviously use a one-dimensional bin packing algorithm to pack segments.

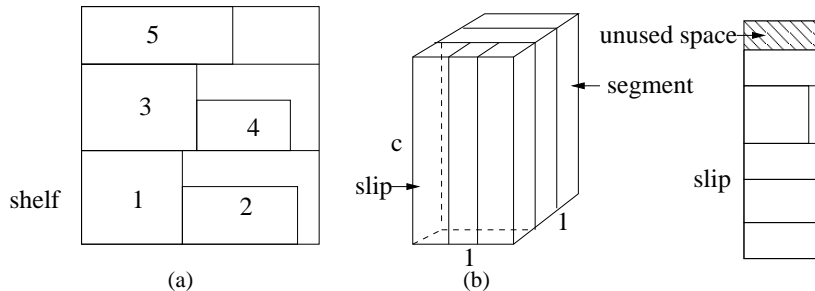


Figure 1: Shelves and segments

**Previous results:** On 3D Strip Packing, Li and Cheng [17] presented the first approximation algorithm with asymptotic worst case ratio 3.25. Two years later, they gave an online algorithm for the problem with asymptotic worst-case (competitive) ratio arbitrarily close to  $(1.69103\dots)^2 \approx 2.8596$  [18]. Then Miyazawa and Wakabayashi [21, 22] improved the asymptotic worst-case ratio to 2.67 and 2.64. Very recently, Jansen and Solis-Oba [14] improved the asymptotic worst-case ratio to  $2 + \epsilon$ .

On 2D Strip Packing, Coffman et al. [6] presented algorithms based on NFDH (Next Fit Decreasing Height) and FFDH (First Fit Decreasing Height), and showed that the respective asymptotic worst-case ratios are 2 and 1.7. Golan [13] and Baker et al. [1] improved the bound to  $4/3$  and  $5/4$ , respectively. Using linear programming and randomization techniques, an asymptotic fully polynomial time approximation schemes (AFPTAS) was given by Kenyon and Rémila [16].

On 2D Bin Packing, in 1982, Chung, Garey and Johnson [5] presented an approximation algorithm with asymptotic worst-case ratio at most 2.125. Caprara [3] improved the upper bound to 1.6910.... On the other hand, Bansal et al. [2] showed that the 2D bin packing problem does not admit an APTAS. Chlebík and Chlebíková [4] further gave an explicit lower bound  $1 + \frac{1}{2196}$ . Since the 2D bin packing problem is a special case of the 3D strip packing problem, the lower bound holds for 3D strip packing too.

## 2 Problems and Notations

Our model is exactly the same as [14]. Given an input list  $L$  of  $n$  three-dimensional boxes, in which each box has length, width and height at most 1 respectively, 3SP is to pack all boxes into a 3D strip (rectangular parallelepiped) of width 1, length 1 and minimum height, so that the boxes do not overlap. In this paper we consider the orthogonal version of the problem without rotations, i.e., the boxes must be packed so that their faces are parallel to the faces of the strip and the boxes are oriented and cannot be rotated. The problem is obviously NP-hard. For

approximation algorithms, we use the standard measure to evaluate them, i.e., the worst-case ratio. In this paper, we consider the asymptotic worst-case ratio. Given an input list  $L$  and an approximation algorithm  $A$ , we denote by  $OPT(L)$  and  $A(L)$ , respectively, the height used by an optimal algorithm and the height by algorithm  $A$  for list  $L$ . The *asymptotic worst-case ratio*  $R_A^\infty$  of algorithm  $A$  is defined by

$$R_A^\infty = \lim_{n \rightarrow \infty} \sup_L \{A(L)/OPT(L) | OPT(L) = n\}.$$

### 3 Basic tools for algorithms and their analysis

**Fractional Bin Packing (FBP).** The continuous version of bin packing plays an important role in designing an asymptotic PTAS [26, 15]. We first give its definition and some properties. Given an instance  $I$  of one dimensional bin packing, suppose that there are  $p$  *distinct* sizes of the items in  $I$ , where  $p$  is a constant. Let  $s_1 > s_2 > \dots > s_p$  be the distinct item sizes and  $n_j$  be the number of items of size  $s_j$  for  $j = 1, \dots, p$ . A *feasible pattern* is a vector  $v = (v_1, \dots, v_p)$  such that  $\sum_{j=1}^p v_j s_j \leq 1$ , i.e., all items in a feasible pattern would fit in one bin. Let  $\nu$  denote the collection of all feasible patterns for  $I$  and  $v^i = (v_1^i, \dots, v_p^i)$  denote the  $i$ -th pattern in  $\nu$ , where  $v_j^i$  is the number of items of size  $s_j$  in the  $i$ -th pattern. We further denote  $x_i$  to be the number of bins being needed for packing the  $i$ -th feasible pattern in  $\nu$ . If we allow  $x_i$  to be a fractional number, then the problem becomes the fractional bin packing problem (FBP) and corresponds to the following *Linear Program* (LP):

$$\begin{aligned} \text{Min} \quad & \sum_{v^i \in \nu} x_i \\ \text{s.t.} \quad & \sum_{v^i \in \nu} v_j^i x_i \geq n_j, \quad j = 1, \dots, p \\ & x_i \geq 0, \quad v^i \in \nu. \end{aligned} \tag{1}$$

The *LP dual* of (1) is given as follows:

$$\begin{aligned} \text{max.} \quad & \sum_{j=1}^p n_j \pi_j \\ \text{s.t.} \quad & \sum_{j=1}^p v_j^i \pi_j \leq 1, \quad v^i \in \nu \\ & \pi_j \geq 0, \quad j = 1, \dots, p. \end{aligned} \tag{2}$$

Optimal values for (1) and (2) coincide and the following important lemma is due to [3],

**Lemma 1** *There exists an optimal solution  $\pi^*$  of (2) such that  $\pi_1^* \geq \pi_2^* \geq \dots \geq \pi_p^*$  (recalling  $s_1 > s_2 > \dots > s_p$ ).*

The following lemma [3, 26], says that the optimal values for BP and FBP are almost equal.

**Lemma 2** *For any bin packing instance  $I$  and for any  $\epsilon > 0$ , we have  $OPT_{BP}(I) \leq (1 + \epsilon)OPT_{FBP}(I) + O(\epsilon^{-2})$ , where  $OPT_{FBP}(I)$  is the optimal value for FBP.*

**Harmonic algorithm.** The Harmonic algorithm was introduced by Lee and Lee [19]. Given a (one-dimensional) bin packing instance  $I$  and an integer  $k > 0$ , we say an item  $i$  belongs to type  $t$  if its size  $s_i \in (\frac{1}{t+1}, \frac{1}{t}]$  for  $t = 1, \dots, k-1$  and to type  $k$  if  $s_i \in (0, \frac{1}{k}]$ , where  $k$  is a constant. Then

the Harmonic algorithm packs items of different types into different bins. During packing, if the current item of type  $t$  does not fit in the corresponding bin, then the algorithm closes the bin and opens a new one. Given an item of size  $x$ , we define a weighting function  $f_k(x)$  as follows:

$$f_k(x) = \begin{cases} \frac{1}{t}, & \text{if } \frac{1}{t+1} < x \leq \frac{1}{t} \text{ with } 1 \leq t < k, \\ \frac{kx}{k-1} & \text{if } 0 < x \leq \frac{1}{k}. \end{cases}$$

Let  $t_1 = 1$ ,  $t_{i+1} = t_i(t_i + 1)$  for  $i \geq 1$ . For a positive integer  $k$ , let  $m(k)$  be the integer such that  $t_{m(k)} < k \leq t_{m(k)+1}$ .  $T_k = \sum_{i=1}^{m(k)} \frac{1}{t_i} + \frac{1}{t_{m(k)+1}} \cdot \frac{k}{k-1}$ . Note that  $T_\infty = \lim_{k \rightarrow \infty} T_k \approx 1.69103$

The weighting function  $f_k(x)$  satisfies the following property (see [19]):

**Lemma 3** For each sequence  $x_1, \dots, x_m$  with  $x_i \in (0, 1]$  and  $\sum_{i=1}^m x_i \leq 1$ ,

$$\sum_{i=1}^m f_k(x_i) \leq T_k.$$

**NFDH packing.** *NFDH* was first proposed by Meir and Moser [20] for packing a set of *squares* into a rectangular bin, but *NFDH* packing can also be applied to pack rectangles. It simply works as follows. First sort all rectangles in non-increasing order of their heights. Then pack them into the bin level by level and in each level we use the Next Fit (NF) algorithm. If a level cannot accommodate the current rectangle, then we close it (will never be used again) and open a new one. (see Figure 2 (c)). Note that NFDH packing can be extended for multidimensional packing [20, 2].

## 4 New upper bounds for 3D strip packing

We call our algorithm 3D Segment Strip Packing (3SSP).

### 4.1 Algorithm 3SSP

Given an item  $R = (l, w, h)$ , where  $l$ ,  $w$  and  $h$  are its length, width and height respectively, we may use  $l(R)$ ,  $w(R)$  and  $h(R)$  to denote the three parameters as well. Algorithm 3SSP has the following 3 main steps.

1. Divide all items into  $k$  groups  $G_1, G_2, \dots, G_k$  such that those in  $G_i$  have their lengths in range  $(\frac{1}{i+1}, \frac{1}{i}]$ , where  $k$  is a constant.
2. Sort all  $G_i$ -items by their width such that  $G_i = (R_1, R_2, \dots, R_{n_i})$  and  $w(R_1) \geq w(R_2) \geq \dots \geq w(R_{n_i})$ , where  $n_i$  is the number of items in group  $G_i$  for  $1 \leq i \leq k$ . Then pack all items in  $G_i$ , for  $1 \leq i < k$ , into segments by algorithm GNF (given later). For  $i = k$ , pack all items in  $G_k$  into segments by algorithm GNFDH (given later).
3. When all items in group  $G_i$  have been packed into segments, for  $1 \leq i \leq k$ , then regard all segments as one dimensional items (Only their width is considered) and call an asymptotic PTAS for one-dimensional bin packing (e.g. [15, 26]) to pack these segments.

In the following we give the procedures to pack 3D items into segments, which are the cores of algorithm 3SSP. We deal with  $G_i$  items ( $1 \leq i < k$ ) and  $G_k$  items separately.

**GNF:** Consider  $G_i$  items ( $1 \leq i < k$ ). Given  $G_i = (R_1, R_2, \dots, R_{n_i})$  such that  $w(R_1) \geq w(R_2) \geq \dots \geq w(R_{n_i})$ .

1. Open a new segment of size  $(1, w_y, c)$ , where  $w_y \leftarrow w(R_1)$ .
2. Divide this segment into  $i$  pieces of slips of sizes  $(\frac{1}{i}, w_y, c)$ , as shown in Figure 2 (a), then without considering their widths and lengths, pack items into these slips by Next Fit. (see Figure 2 (b)).
3. If there are remaining items, re-index them and go to Step 1.

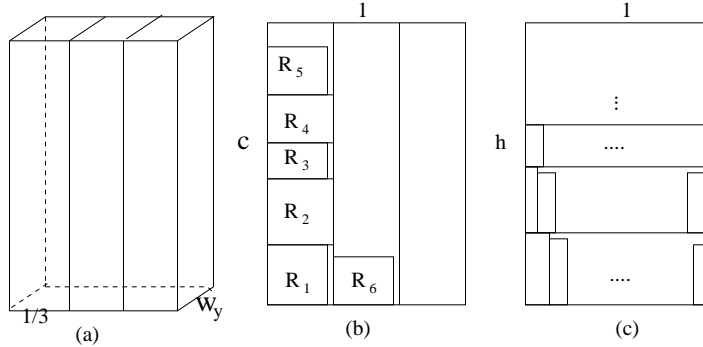


Figure 2: GNF for  $G_3$ ; the projection of GNFDH packing without considering widths

**GNFDH:** Given  $G_k = (R_1, R_2, \dots, R_{n_k})$  such that  $w(R_1) \geq w(R_2) \geq \dots \geq w(R_{n_k})$ .

1. Open a new segment with size  $(1, w_y, c)$ , where  $w_y \leftarrow w(R_1)$ .
2. Find a maximal index  $j$  such that  $R_1, R_2, \dots, R_j$  can be placed into the segment by NFDH without considering their widths. Pack the  $j$  items by NFDH. (See Figure 2 (c))
3. Re-index the remaining items in  $G_k$  (if any), go to Step 1.

## 4.2 Analysis of the algorithm

In the algorithm analysis, *dual feasible functions* by Fekete and Scheper [9] play a crucial role. (Similar notions are used as weighting functions [11, 12, 19, 24, 25]) Suppose that a function  $f : [0, 1] \rightarrow [0, 1]$  satisfies  $\sum_{i=1}^m f(x_i) \leq 1$  for any sequence  $x_1, \dots, x_m$  such that  $\sum_{i=1}^m x_i \leq 1$  and  $x_i \in [0, 1]$ . Then  $f$  is called a *dual feasible function*. Here are two specific examples: Let  $\bar{\pi} = (\bar{\pi}_1, \dots, \bar{\pi}_p)$  be a feasible solution of (2) (dual LP for FBP in Section 3) satisfying the requirement of Lemma 1 and let  $\bar{\pi}_{p+1} := 0$ ,  $s_0 := 1$  and  $s_{p+1} := 0$ . Define a new function  $g$  by

$$g(0) = 0, \text{ and } g(x) = \bar{\pi}_j, \text{ for } x \in [s_j, s_{j-1}).$$

The other example is  $f_k$  defined in Section 3.

**Lemma 4** [3] Both  $g(x)$  and  $\frac{f_k(x)}{T_k}$  are dual feasible functions.

Using these two functions, we define the *modified volume*  $W(R)$  of an item  $R = (l, w, h)$  as

$$W(R) = f_k(l) \cdot g(w) \cdot h.$$

The total modified volume of the input list  $L$  of items is  $W(L) = \sum_{R \in L} W(R)$ .

We need one more lemma regarding dual feasible functions and 2D packing: let  $(l_1, w_1), \dots, (l_m, w_m)$  be 2D items which can be packed into a square bin of size  $(1,1)$ , and  $f_1$  and  $f_2$  be dual feasible functions. Then we have the following lemma (see [7, 25] for the proof), which is important for bounding the total modified volume.

**Lemma 5**  $\sum_{i=1}^m f_1(l_i) f_2(w_i) \leq 1$ .

Now, we are ready to prove the upper bound for the approximability of our algorithm 3SSP. Let  $I(L)$  be the 1-dimensional item list obtained after Step 2 of 3SSP, i.e., the list of the widths of the segments. Recall that  $c$  is the height of the segment and  $k$  is the parameter of the Harmonic algorithm. Let  $OPT_{BP}(I(L))$  be the optimal cost of 1-dimensional bin packing for the list  $I(L)$  and  $OPT(L)$  be the optimal cost for 3D Strip Packing for the list  $L$ . Our goal is thus to prove the following theorem.

**Theorem 1** For any  $\epsilon > 0$ ,  $c \cdot OPT_{BP}(I(L)) \leq \frac{c}{c-1}(1 + \epsilon)T_k OPT(L) + O(ck\epsilon^{-2})$ .

Since we employ some APTAS for packing  $I(L)$ , algorithm 3SSP achieves the cost arbitrarily close to  $c \cdot OPT_{BP}(I(L))$  in the asymptotic case. It shows that the asymptotic worst-case ratio of 3SSP is at most  $\frac{c}{c-1}(1 + \epsilon)T_k$  for any given  $\epsilon > 0$ , which tends to  $T_\infty$  as  $\epsilon \rightarrow 0$  and the constants  $c$  and  $k$  take sufficiently large integers.

The basic idea of the proof is to establish the relation of the left and right-hand sides of the inequalities in the theorem to the total modified volume. Recall that 3SSP uses different segments for each  $G_i$ . A segment is called type  $i$  if it contains  $G_i$  items. For  $q = 1, \dots, k$ , let  $m^q$  be the number of segments of type  $q$  and  $w_i^q$  the width of the  $i$ -th segment of type  $q$ , where  $1 \leq i \leq m^q$ . By algorithm 3SSP, we have

$$w_1^q \geq w_2^q \geq \dots \geq w_{m^q}^q. \quad (3)$$

Noting that  $g(\cdot)$  is the function defined in Subsection 4.2 for instance  $I(L)$ , by Lemma 1, we have

$$g(w_1^q) \geq g(w_2^q) \geq \dots \geq g(w_{m^q}^q). \quad (4)$$

Let  $G^q := \sum_{i=1}^{m^q} g(w_i^q)$  denote the total modified width of the segments of type  $q$ . Now, we give a lower bound for  $W(L)$ . For convenience, we define  $w_{m^q+1}^q = 0$  for all  $q$ 's.

**Lemma 6** The total modified volume  $W(L) > (c-1) \sum_{q=1}^k G^q - ck$ .

*Proof.* Let  $S_i^q$  be the  $i$ -th segment of type  $q$  and  $L_i^q$  be the set of all boxes in  $S_i^q$ . We first prove that

$$W(L_i^q) \geq (c-1)g(w_{i+1}^q). \quad (5)$$

Case 1.  $q \neq k$ . If  $i = m^q$ ,  $W(L_i^q) \geq (c-1)g(w_{i+1}^q)$  since  $g(w_{m^q+1}^q) = 0$  and  $W(L_{m^q}^q) \geq 0$ . Otherwise, i.e.,  $1 \leq i < m^q$ , by GNF packing, we have the height packed in every slip of  $S_i^q$  is at least  $(c-1)$  and every box in  $S_i^q$  has length in  $(\frac{1}{q+1}, \frac{1}{q}]$  and width at least  $w_{i+1}^q$  (i.e., the width of the next segment). Remember  $f_k(x) = \frac{1}{q}$  where  $x$  in  $(\frac{1}{q+1}, \frac{1}{q}]$ . By (4), the total weight of the boxes in every slip is at least

$$\frac{1}{q} \cdot g(w_{i+1}^q) \cdot (c-1) = \frac{c-1}{q} \cdot g(w_{i+1}^q).$$

Since there are  $q$  slips in segment  $S_i^q$ ,

$$W(L_i^q) \geq q \cdot \frac{c-1}{q} \cdot g(w_{i+1}^q) = (c-1)g(w_{i+1}^q).$$

Case 2.  $q = k$ . If  $i = m^q$ ,  $W(L_i^q) \geq (c-1)g(w_{i+1}^q)$  still holds. Otherwise, i.e.,  $1 \leq i < m^q$ . Consider GNFDH packing for items of type  $k$ . Assume there are  $l$  levels in  $S_i^k$ , and their heights are  $h_1, h_2, \dots, h_l$ , respectively. Set  $h_{l+1} = c - \sum_{j=1}^l h_j$ . By NFDH packing, if  $i < m^q$ , we have  $h_1 \geq h_2 \geq \dots \geq h_{l+1}$  and every box in  $S_i^k$  has width at least  $w_{i+1}^k$ , length at most  $\frac{1}{k}$ . Hence the total sum of lengths in every level is at least  $1 - \frac{1}{k}$ . Remember  $f_k(x) = \frac{kx}{k-1}$  where  $x$  in  $[0, \frac{1}{k}]$ . And every box in the  $j$ -th level of  $S_i^k$  has height at least  $h_{j+1}$ . So the total weight in the  $j$ -th level is at least

$$h_{j+1} \cdot g(w_{i+1}^k) \sum_R \frac{k}{k-1} l(R) \geq h_{j+1} \cdot g(w_{i+1}^k).$$

Since  $\sum_{j=1}^l h_j \geq c-1$ , the total weight of  $L_i^k$  is

$$W(L_i^k) \geq \sum_{j=1}^l g(w_{i+1}^k) h_j \geq (c-1)g(w_{i+1}^k).$$

So the inequality (5) holds.

Since there are  $k$  types of segments and in every type  $q$  there are  $m^q$  segments, so

$$W(L) = \sum_{q=1}^k \sum_{i=1}^{m^q} W(L_i^q).$$

By (5),

$$\begin{aligned} W(L) &\geq (c-1) \sum_{q=1}^k \sum_{i=1}^{m^q} g(w_{i+1}^q) \\ &= (c-1) \sum_{q=1}^k (G^q - g(w_1^q)) \\ &> (c-1) \sum_{q=1}^k G^q - ck. \end{aligned}$$

The last inequality follows directly from  $g(w_1^q) \leq 1$ . □

Next we give an upper bound for total modified volume  $W(L)$ .

**Lemma 7** *Given any input list  $L$  over  $[0, 1]^3$ , the total modified volume  $W(L) \leq T_k \text{OPT}(L)$ .*

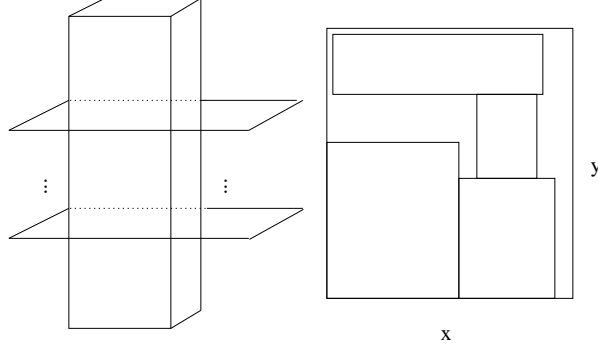


Figure 3: Cutting an optimal packing and a layer projection on  $xy$ -plane

*Proof.* Consider an optimal packing for an input list  $L$ . For each item of  $L$  we draw two horizontal planes at its bottom and top, shown as Figure 3. These planes cut the optimal packing into layers such that all items (may be part of the original items) in a layer have the same height. Then we can see that each layer is associated with a *feasible packing* on a square bin of  $(1, 1)$  by ignoring the heights. Assume that after cutting, totally, there are  $l$  layers and their heights are  $\delta_1, \delta_2, \dots, \delta_l$ , respectively. By Lemmas 3 and 4, we have  $\sum_{x \in S} f_k(x) \leq T_k$  and  $\sum_{x \in S} g(x) \leq 1$  for any list  $S$  with  $\sum_{x \in S} x \leq 1$ . Since in the  $i$ -th layer, every item has height  $\delta_i$ , by Lemma 5, the total weight of all items in the  $i$ -th layer is at most

$$T_k \times 1 \times \delta_i.$$

Since

$$\sum_{i=1}^l \delta_i = OPT(L),$$

then

$$W(L) \leq \sum_{i=1}^l \delta_i T_k = T_k \cdot OPT(L).$$

□

Now it is straightforward to prove Theorem 1.

(Proof of Theorem 1) By Lemma 2, we have

$$OPT_{BP}(I(L)) \leq (1 + \epsilon)OPT_{FBP}(I(L)) + O(\epsilon^{-2}).$$

By the duality of FBP and the dual FBP, as used in [3], we have

$$OPT_{FBP}(I(L)) = \sum_{q=1}^k G^q,$$

where  $G^q := \sum_{i=1}^{m^q} g(w_i^q)$  denotes the overall modified width of the segments of type  $q$ . By Lemmas 6 and 7,

$$c \cdot OPT_{BP}(I(L)) \leq c(1 + \epsilon)OPT_{FBP}(I(L)) + O(c\epsilon^{-2})$$



$$\begin{aligned}
&= c(1 + \epsilon) \sum_{q=1}^k G^q + O(c\epsilon^{-2}) \\
&< c(1 + \epsilon) \frac{W(L) + ck}{c - 1} + O(c\epsilon^{-2}) \\
&\leq \frac{c}{c - 1} (1 + \epsilon) T_k OPT(L) + O(ck\epsilon^{-2}).
\end{aligned}$$

□

**Remark.** Our algorithm can also be applied to the parametric case in which the boxes have bounded length (or width). Then by Theorem 1 the asymptotic worst-case ratio in the parametric case that all boxes have width or length bounded from above by  $\alpha$  is stated in the following table, which is better than the previous parametric ratio  $R_{para}^\infty$  in [23].

$\alpha \in$	$(\frac{1}{2}, 1]$	$(\frac{1}{3}, \frac{1}{2}]$	$(\frac{1}{4}, \frac{1}{3}]$	$(\frac{1}{5}, \frac{1}{4}]$
$R_{3SSP}^\infty$	1.691...	1.423...	1.302...	1.234...
$R_{para}^\infty$	3.050...	2.028...	1.684...	1.511...

## 5 APTAS for packing items with square bases

In this section, by combining the techniques for 2D strip packing [16] and 2D bin packing [2], we give an APTAS for the case that the boxes have square bases (bottoms).

The standard ideas in our scheme are below:

- Create a gap between large items and small items such that the items fall into the gap do not affect the packing significantly.
- Pack large items in the way similar to 2D strip packing [16] and pack the other items by NFDH [20, 2, 14].

We use a multidimensional version of NFDH in [20, 2], called MNFDH, to pack items with small base sizes into a 3D bin or a strip. The lemma below can be obtained directly from [20] (see also [2, 14]).

**Lemma 8** *Let  $I$  be a set of 3D boxes with base sides at most  $\delta$  and height at most 1. Consider the MNFDH packing applied to  $I$ . If MNFDH cannot place more boxes from  $I$  into a bin of size  $(a, b, c)$ , then either all boxes of  $I$  has been packed into the bin or the total packed volume in the bin is at least  $(a - \delta)(b - \delta)(c - 1)$ .*

Given any feasible 3D strip packing of height  $h$ , we can scan a plane parallel to the square base of the strip from the bottom to the top to obtain a vector  $x = (x_1, \dots, x_q)$  such that  $\sum_{i=1}^q x_i = h$ , where  $q$  is the number of patterns to pack all squares induced from the input list into a unit square bin and  $x_i$  is the height of pattern  $i$ .

**Definition of  $S(K, \delta)$ .** If an input set  $I$  has a constant number of different sizes, say  $K$ , and all the base sides are at least  $\delta$ , where  $\delta$  is a constant, then we define this problem as *Restricted 3D strip packing with square bases*, denoted by  $S(K, \delta)$ .

**Lemma 9** [2] *The number of all feasible patterns of packing the square items, induced from an instance of  $S(K, \delta)$ , into a unit square bin is a constant.*

**Lemma 10**  $S(K, \delta)$  can be solved within  $OPT + K$  in polynomial time of  $n$ , where  $OPT$  is the optimal cost for  $S(K, \delta)$  and  $n$  is the input size.

The proof is put to the appendix.

**Lemma 11** Assume the input set  $I$  contains boxes with base sides at least  $\delta$ . Then for any  $K > 0$ , we can get a solution within  $OPT(I)(1 + \frac{1}{\delta^2 K}) + K$  in polynomial time for packing  $I$  into the strip.

The proof is put to the appendix.

**Asymptotic PTAS** Using the similar techniques as in [2], we present an APTAS. Given an input set  $I$  and any  $\epsilon > 0$ , our packing is as follows.

1. Let  $w_j$  be the base side length of item  $j$ . Define  $M_i = \{j : w_j \in [\epsilon^{2^{i+1}-1}, \epsilon^{2^i-1}]\}$  for  $i = 1, \dots, r + 1$ , where  $r = \lceil 1/\epsilon \rceil$ .
2. Set  $M := M_i$  for some index  $1 \leq i \leq r$  satisfying  $Vol(M_i) \leq \epsilon Vol(I)$  (such a set  $M_i$  must exist), where  $Vol(X)$  is the total volume of items in  $X$ . Define the set of large items as  $L = \{j : w_j \geq \epsilon^{2^i-1}\}$  and the set of small items as  $S = \{j : w_j < \epsilon^{2^{i+1}-1}\}$ .
3. Set  $K = \lceil 1/(\epsilon \delta^2) \rceil$  and round all items in  $L$  up into  $K$  distinct sizes,  $\delta = \epsilon^{2^i-1}$ . Then call the algorithm in Lemma 11 to get an almost optimal solution.
4. Partition the unused space in the current strip into cuboid regions and use MNFDH to pack as many squares in  $S$  as possible into the free space. Let  $S' \subset S$  denote the subset of the remaining small items that could not be packed ( $S'$  could possibly be empty).
5. Use MNFDH to pack  $M \cup S'$  at the top of the current packing in the strip.

**Theorem 2** Given an input set  $I$  of 3D boxes with square bases,  $A(I) \leq (1 + 12\epsilon)OPT(I) + O(K)$ , where  $A(I)$  is the height used by our algorithm and  $K = \epsilon^{-O(2^{\epsilon^{-1}})}$ .

*Proof.* (Sketch.) Our argument is similar as [2]. After Step 4, there are two cases.

Case 1.  $S'$  is not empty. Then by the proof in Section 3.4 of [2],

$$A(I) \leq Vol(I)/(1 - 6\epsilon) + O(K) \leq (1 + 12\epsilon)OPT(I) + O(K).$$

The last step follows by assuming without loss of generality that  $\epsilon \leq 1/12$ .

Case 2.  $S'$  is empty. Set  $K = 1/(\epsilon \delta^2) = \epsilon^{-O(2^{\epsilon^{-1}})}$ , where  $\delta = \epsilon^{2^i-1}$  in Step 2. By Lemma 11,

$$A(L \cup S) = A(L) \leq (1 + \frac{1}{\delta^2 K})OPT(I) + K \leq (1 + \epsilon)OPT(I) + O(\epsilon^{-O(2^{\epsilon^{-1}})}). \quad (6)$$

Next, we consider the cost of packing  $M$  by MNFDH. Since the base size of each item in  $M$  is at most  $\epsilon$ , by Lemma 8,

$$A(M) \leq Vol(M)/(1 - 2\epsilon) + 1 \leq \epsilon OPT(I)/(1 - 2\epsilon) + 1. \quad (7)$$

Combining (6) and (7),  $A(I) \leq (1 + 3\epsilon)OPT(I) + O(\epsilon^{-O(2^{\epsilon^{-1}})})$ .

Finally we want to note that each step in our algorithm takes polynomial time of  $n$  since  $\epsilon$  is a constant.  $\square$

## 6 Conclusions

In this paper, we present new asymptotic upper bounds for the 3D strip packing problems. Our results give a possible way to apply the approaches for 1- and 2-dimensional bin packing to 3-dimensional strip packing. It might be interesting to see if the idea can be used to tackle higher dimensional strip packing in the general case. Regarding the special case that items have square bases, with the technique in the previous work on 2D bin packing and 2D strip packing an APTAS is easily achieved. Such an approach can also be extended to multidimensional strip packing.

## References

- [1] B.S. Baker, D.J. Brown, and H.P. Katseff, A  $5/4$  algorithm for two-dimensional packing. *J. Algorithms* 2:348-368, 1981.
- [2] N. Bansal, J.R. Correa, C. Kenyon, and M. Sviridenko, Bin packing in multiple dimensions: Inapproximability results and approximation schemes, *Mathematics of Operations Research* 31(1):31-49, 2006.
- [3] A. Caprara, Packing 2-dimensional bins in harmony, *FOCS* 2002: 490-499.
- [4] M. Chlebík, J. Chlebíková, Inapproximability results for orthogonal rectangle packing problems with rotations, *CIAC* 2006: 199-210.
- [5] F.R.K. Chung, M.R. Garey and D.S. Johnson, On packing two-dimensional bins, *SIAM Journal on algebraic and discrete methods* 3:66-76, 1982.
- [6] E.G. Coffman, M.R. Garey, D.S. Johnson, and R.E. Tarjan, Performance bounds for level oriented two dimensional packing algorithms, *SIAM Journal on Computing* 9:808-826, 1980.
- [7] J. Csirik and A. van Vliet, An on-line algorithm for multidimensional bin packing, *Operations Research Letters* 13(3):149-158, 1993.
- [8] J. Csirik and G.J. Woeginger, Shelf Algorithms for On-Line Strip Packing, *Information Processing Letters* 63(4):171-175, 1997.
- [9] S.P. Fekete and J. Schepers, New classes of fast lower bounds for bin packing problems, *Mathematical Programming* 91:11-31, 2001.
- [10] P.C. Gilmore and R.E. Gomory, Multistage cutting problems of two and more dimensions, *Operations Research* 13:94-119, 1965.
- [11] M.R. Garey, R.L. Graham, D.S. Johnson, and A.C. Yao, Resource constrained scheduling as generalized bin packing, *J. Comb. Th. Ser. A* 21:257-298, 1976.
- [12] M.R. Garey, R.L. Graham, and J.D. Ullman, Worst-case analysis of memory allocation algorithms, *STOC* 1972: 143-150.
- [13] I. Golan, Performance bounds for orthogonal, oriented two-dimensional packing algorithms, *SIAM Journal on Computing* 10: 571-582, 1981.

- [14] K. Jansen and R. Solis-Oba, An asymptotic approximation algorithm for 3D-strip packing, *SODA* 2006: 143-152.
- [15] N. Karmarkar and R.M. Karp, An efficient approximation scheme for the one-dimensional bin-packing problem, *FOCS* 1982: 312-320.
- [16] C. Kenyon and E. Rémila, A near-optimal solution to a two-dimensional cutting stock problem, *Mathematics of Operations Research* 25:645-656, 2000.
- [17] K. Li and K.H. Cheng, On three-dimensional packing, *SIAM Journal on Computing* 19:847-867, 1990.
- [18] K. Li and K.H. Cheng, Heuristic algorithms for on-line packing in three dimensions, *Journal of Algorithms* 13:589-605, 1992.
- [19] C.C. Lee and D.T. Lee, A simple on-line packing algorithm, *J. ACM* 32:562-572, 1985.
- [20] A. Meir and L. Moser, On packing of squares and cubes, *Journal of combinatorial theory* 5:126-134, 1968.
- [21] F.K. Miyazawa and Y. Wakabayashi, An algorithm for the three-dimensional packing problem with asymptotic performance analysis, *Algorithmica* 18:122-144, 1997.
- [22] F.K. Miyazawa and Y. Wakabayashi, Packing problems with orthogonal rotations, *Proceedings of the 6th Latin American Symposium on Theoretical Informatics*, LNCS 2976, 359-368, 2004.
- [23] F.K. Miyazawa and Y. Wakabayashi, Two- and three-dimensional parametric packing, *Computers and Operations Research*, to appear.
- [24] S.S. Seiden, On the online bin packing problem, *J. ACM* 49(5):640-671, 2002.
- [25] S.S. Seiden and R. van Stee, New bounds for multidimensional packing, *Algorithmica* 36:261-293, 2003.
- [26] Fernandez de la Vega and G.S. Lueker, Bin packing can be solved within  $1 + \epsilon$  in linear time, *Combinatorica* 1(4):349-355, 1981.

## Appendix

### Proof of Lemma 10

*Proof.* Our idea is similar with the one in [16]. First we consider the following LP, where  $q$  is the number of all feasible patterns of packing the squares induced from an instance of  $S(K, \delta)$  into a unit square bin,  $\alpha_{ij}$  is the number of type  $j$  items in pattern  $i$  and  $\beta_j$  is the sum of heights of type  $j$  items for  $1 \leq j \leq K$ ,  $x_i$  is the height of pattern  $i$ .

$$\begin{array}{ll} \text{Min.} & \sum_{i=1}^q x_i \\ \text{s.t.} & \sum_{i=1}^q \alpha_{ij} x_i \geq \beta_j, \quad 1 \leq j \leq K \\ & x_i \geq 0, \quad 1 \leq i \leq q. \end{array}$$

By Lemma 9,  $q$  is a constant related to  $K$  and  $\delta$ . So, the above LP can be solved in polynomial time of  $n$ , where  $n$  is the input instance size of  $S(K, \delta)$ . Let  $x^* = (x_1^*, \dots, x_q^*)$  be an optimal solution of the above LP. By some linear programming property, there are at most  $K$  non-zero  $x_i^*$ 's. Up to renaming, we assume the non-zero coordinates are  $x_1^*, \dots, x_K^*$ . We construct a packing of  $S(K, \delta)$  in the following way.

We fill in the strip bottom-up, taking each pattern in turn. Let  $x_j^* > 0$  be the current pattern. Pattern  $j$  will be used between level  $l_j = (x_1^* + 1) + \dots + (x_{j-1}^* + 1)$  and level  $l_{j+1} = l_j + x_j^* + 1$  (initially  $l_1 = 0$ ). For each  $i$  such that  $\alpha_{ij} \neq 0$ , we draw  $\alpha_{ij}$  cuboids of base size  $w_i$  going from level  $l_j$  to level  $l_{j+1}$ , where  $w_i$  is the base side length of type  $i$  item. After this is done for all  $j$ 's, we take all the cuboids of width  $w_i$  one by one in some arbitrary order, and fill them in with the boxes of base size  $w_i$  in a greedy manner (some small amount of space may be wasted on top of each column).

Since every box has its height at most 1, all boxes can be packed in the above way. Moreover  $\sum x_i^*$  is a lower bound of the optimal value for  $S(K, \delta)$ . Hence we have this lemma.  $\square$

### Proof of Lemma 11

*Proof.* The algorithm has 3 steps:

**Stacking.** Sort the  $n$  boxes in non-increasing order of base sizes and stack up them one by one to get a stack of height  $H$ . And define  $K - 1$  threshold boxes, where a box is a threshold if its interior or bottom base intersects some plane  $z = \frac{iH}{K}$ , for  $1 \leq i \leq K - 1$ .

**Grouping and rounding.** The threshold boxes divide the remaining boxes into  $K$  groups. The base sizes of the boxes in the first group are rounded up to 1, and the base sizes of the boxes in each subsequent group are rounded to the base size of the threshold box below their group. This defines an instance  $I_{sup}$  of  $S(K, \delta)$ .

**Packing.** Apply the approach of Lemma 10 to  $I_{sup}$  and output the packing.

To analyze the algorithm, we use the argument of Kenyon and Rémila [16]. Consider two instances  $I'_{inf}$  and  $I'_{sup}$  derived from the stack built in stacking step. The two instances are obtained by first cutting the threshold boxes using the planes  $z = \frac{iH}{K}$ , then considering the  $K$  subsequent groups of boxes in turn (where each group now has cumulative height exactly  $H/K$ ); to define  $I'_{sup}$ , we round the base sizes in each group up to the largest base size of the group (up

to 1 for the first group); to define  $I'_{inf}$ , we round the base sizes in each group down to the largest base size of the next group (down to 0 for the last group). Let  $lin(I)$  be the solution of the above linear programming for instance  $I$ . It is easy to see that

$$lin(I'_{inf}) \leq lin(I) \leq lin(I_{sup}) \leq lin(I'_{sup}).$$

Moreover,

$$lin(I'_{sup}) \leq lin(I'_{inf}) + H/K.$$

Since  $OPT(I) \geq lin(I)$  and  $OPT(I) \geq \delta^2 H$ , the height used by our packing is at most  $lin(I_{sup}) + K \leq lin(I'_{inf}) + H/K + K \leq OPT(I) + K + H/K \leq OPT(I)(1 + \frac{1}{\delta^2 K}) + K$ .  $\square$